

혼합모델 조립라인의 다목적 투입순서 문제를 위한 유전알고리즘

A Genetic Algorithm for a Multiple Objective Sequencing Problem in Mixed Model Assembly Lines

현철주* · 김여근**

Chul-Ju Hyun* · Yeo-Keun Kim**

Abstract

This paper is concerned with a sequencing problem in mixed model assembly lines, which is important to efficient utilization of the lines. In the problem, we deal with the two objectives of minimizing the risk of stoppage and leveling part usage, and consider sequence-dependent setup time. In this paper, we present a genetic algorithm(GA) suitable for the multi-objective optimization problem. The aim of multi-objective optimization problems is to find all possible non-dominated solutions. The proposed algorithm is compared with existing multi-objective GAs such as vector evaluated GA, Pareto GA, and niched Pareto GA. The results show that our algorithm outperforms the compared algorithms in finding good solutions and diverse non-dominated solutions.

1. 서론

혼합모델 조립라인의 효율적인 이용을 위해서는 라인밸런싱과 모델의 투입순서 문제가 중요하게 다루어져야 한다[3, 18]. 라인밸

런싱은 작업장에 특정목적에 맞도록 작업을 할당하는 문제이며, 투입순서 문제는 조립라인에 투입되는 모델의 생산순서를 결정하는 문제이다. 본 연구는 모델의 투입순서 문제를 다루고 있다.

* 정읍공업전문대학 품질관리과

** 전남대학교 공과대학 산업공학과

혼합모델 조립라인의 투입순서에 관한 연구는 크게 조립라인의 효율적인 이용을 위한 투입순서와 모든 부품의 사용률을 일정하게 유지하기 위한 투입순서로 나누어 이루어져 왔다[4]. 라인의 효율적인 이용을 위해 컨베이어 정지위험의 최소화[18], 라인길이의 최소화[3], 총 가외작업(Utility Work)의 최소화[22], 작업준비비용의 최소화[6] 등 다양한 목적들에 대해 연구되었으며, Just In Time(JIT) 생산시스템에서 부품사용의 일정률을 유지하기 위한 수리 모형과 발견적기법이 개발되어 왔다[16, 17].

혼합모델 조립라인에서 단 하나의 목적을 사용하여 구한 투입순서는 다른 목적에 대해 상충될 수 있다. 예를 들어 컨베이어 정지위험을 최소화 하는 투입순서는 조립라인에서 사용되는 부품의 일정률을 유지하기 어려울 수 있으며, 라인 길이와 준비시간(또는 비용)의 최소화는 서로 상충되는 목적일 수 있다. 따라서 여러 개의 목적을 동시에 고려해야 할 필요성이 있다. 본 연구에서는 JIT 생산시스템에서 투입순서의 목적으로 흔히 사용되는 컨베이어 정지위험의 최소화와 부품 사용의 일정률 유지의 두 가지 목적을 동시에 다루고자 한다.

혼합모델 조립라인의 대부분 작업장에서는 흔히 조립부품준비, Jig 또는 치공구의 교체 등에 따른 작업준비시간이 요구된다. 컨베이어 정지위험을 분석하는 데 있어서 준비작업은 흔히 컨베이어 밖에서 이루어지므로 컨베이어 상에서 이루어지는 조립작업과는 구별되어야 한다. 준비시간은 모델의 변화에 상관없이 일정한 경우와 모델의 투입순서에 따라 달라지는 경우가 있다. 모델의 투입순서

에 따라 준비시간이 달라지는 한 예로 KA 자동차회사의 승용차 생산라인의 엔진작업장과 변속기 장착작업장을 들 수 있다. 엔진작업장에서는 모델이 변경되면 Jig와 공구를 교체하여야 하는데 모델 A에서 모델 B로 변경되면 Jig에서 모델A에 사용되는 받침용 블록(Block) 2개를 해체하고 B에 사용되는 블록 4개를 설치해야 하고, 조립에 필요한 공구도 교체되어야 한다. 또한 모델 B에서 모델 A로 교체되면 역으로 모델 B에서 사용되는 블록 4개를 해체하고 모델 A용 블록 2개를 설치해야 하며, 공구 또한 교체되어야 한다. 이러한 준비시간은 각각 다르며, 이 시간들은 각각 사이클 타임의 1/4과 1/5정도의 시간이 요구된다. 또한 변속기 장착작업장의 경우 자동변속기에서 수동변속기로의 모델변경과 그 역으로의 모델변경에도 각각 치공구 교체에 따른 준비시간이 필요하게 된다. 따라서 본 연구에서는 모델의 투입순서에 따라 준비시간이 변하는 모형을 다룬다. 이 모형은 각 작업장에서 준비시간이 일정한 경우와 없는 경우의 모형을 포함하고 있다.

다목적 문제를 풀기 위해 가중치를 이용하여 목적함수를 결합하거나 하나의 목적함수를 제외한 나머지 목적함수를 제약식으로 두고 접근하는 방법을 흔히 사용하고 있다. 그러나 이러한 접근방법은 각 목적함수의 중요도에 대한 가중치를 결정하기가 어렵다는 단점을 갖고 있다[19]. 다목적 문제에서는 의사결정자가 자신의 선호도에 따라 해를 선택할 수 있도록 해야 한다. 이를 위해 파레토 최적해를 구할 수 있는 탐색적기법의 개발이 중요하다. 파레토 최적해는 모든 목적에 대해 더 좋은 다른 해가 존재하지 않는 해이다.

혼합모델 조립라인의 투입순서 문제는 조합최적화 문제로 NP-hard이다[21]. 또한 모델의 수요변화, 부품의 결품으로 인한 투입순서의 재결정에는 실시간 처리가 요청된다. 본 연구의 목적은 모델변화에 따른 준비시간을 고려한 컨베이어 정지위험 최소화와 부품사용의 일정률 유지의 목적을 동시에 고려하는 효율적인 다목적 유전알고리즘(Multi-Objective Genetic Algorithm)을 제시하는 데 있다.

유전알고리즘은 일종의 인공지능 기법으로 자연의 진화과정을 모방하여 개발한 탐색 기법이다. 유전알고리즘은 다목적 문제를 해결하는 데 적합한 기법으로 이에 관한 여러 연구[8, 19]가 이루어져 왔다. 유전알고리즘의 가장 큰 특징은 고전적 최적화기법이나 Simulated Annealing, Tabu Search에서와 같이 하나의 해를 이동시키면서 해공간을 탐색하는 것이 아니고, 여러 해(해집단)를 이동시키면서 해공간을 탐색할 수 있다는 점이다. 이러한 여러 해의 유지는 다목적 문제에서 다양한 비지배 해(Non-dominated Solutions)를 구할 수 있게 한다. 다목적 문제에서 하나의 해는 여러 목적함수의 값을 갖기 때문에 유전알고리즘에서 다음 세대에 살아남는 개체를 선택하는 선별방법(Selection Method)을 구현하는 데 어려움이 있다. 따라서 다목적 유전알고리즘에서는 주로 개체의 선별방법에 관하여 연구가 이루어 왔다[10, 12, 20].

본 연구에서는 다목적 투입순서 문제의 다양한 파레토 최적해를 찾기 위해 비지배 순위(Non-domination Rank)와 適所(Niche)의 개념을 도입한 순위-적소 유전알고리즘을 제안하고, 실험을 통하여 순위-적소 유전알고리즘의 성능을 분석하고자 한다.

본 연구의 구성은 다음과 같다. 제 2장에서는 다루는 혼합모델 조립라인의 투입순서 문제를 기술하고, 제 3장에서는 기존의 다목적 유전알고리즘에 대해 살펴본 후, 순위-적소 유전알고리즘을 제안한다. 그리고 제 4장에서는 실험을 통하여 순위-적소 유전알고리즘의 성능을 비교 분석한다. 제 5장은 결론으로 구성되어 있다.

2. 혼합모델 조립라인의 투입순서

2.1 관련연구

혼합모델 조립라인의 투입순서 문제는 Kilbridge와 Wester[13]에 의해 처음으로 제시된 이후, 많은 수리모형과 발견적기법이 개발되었다. Okamura와 Yamashina[18]는 컨베이어의 정지 위험을 최소화 하는 투입순서를 위한 발견적기법에 관한 연구로, 모델의 생산순서를 상호 교환하거나 삽입하는 방법을 제시하고 있다. Tsai[21]는 투입순서 문제가 NP-hard임을 보이고, 두 가지 모델과 단일 작업장의 상황에서 라인정지 위험과 총가외작업의 최소화를 위한 최적해를 찾는 알고리즘을 제시하고 있다. Monden[17]은 부품 사용의 일정률을 유지하고자, Toyota에서 사용하는 목표추적법(Goal-Chasing Method)을 제시하였고, Miltenburg[16]는 JIT생산시스템하에서 단위시간당 부품의 사용률을 가능하면 일정하게 유지하는 투입순서를 구하는 수리모형과 발견적 기법을 제시하였다. 또한 Bard *et al.* [4]은 라인길이의 최소화과 부품사용의 일정률을 유지하는 투입순서를 위한 수리모형을 제시하였다. 이들은 가중치를 사용하여 목적함수를 결합하고, Tabu Search를 적용하였다.

김여근과 현철주[14]는 라인길이를 최소로 하는 투입순서를 구하기 위한 유전알고리즘을 제시하였다. 한편 준비비용을 최소로 하기 위해, Burns와 Daganzo[6]는 모델변화에 따른 준비비용과 생산능력을 고려하여 투입순서를 결정하는 방법을 제시하였으며, Bolat et al. [5]은 준비비용과 가외작업비용(Utility Work Cost)을 최소로 하는 수리모형과 발전적 기법을 제시하였다.

2.2 문제의 설정

본 연구에서 다루는 혼합모델 조립라인은 일정한 속도(V_c)로 이동하는 컨베이어 시스템으로 유사한 여러 모델의 제품들이 일정시간 간격으로 투입되고 있으며, 컨베이어의 이동에 따라 작업자가 이동하면서 작업하는 라인이다. 라인은 J 개의 폐쇄작업장으로 구성되어 있고, 한 사람의 작업자로 운영된다. 폐쇄작업장은 작업을 위한 설비나 장비의 제약으로 인한 경계가 존재하여 작업 지연으로 후행 작업장에서 작업을 하거나 선행 작업장에서 미리 작업하는 것이 불가능한 작업장이다. 작업자의 이동 시간은 무시되며, 각 작업장은 라인밸런싱되었다고 가정한다.

혼합모델 조립라인의 설계시 고려하는 요소로는 작업자의 작업계획(Operator Schedule), 사이클당 생산제품의 수, 제품의 투입시간 간격 등이 있다[3]. 첫째, 작업자의 작업을 계획하는 방법에는 늦은 작업계획(Late Start Schedule)과 빠른 작업계획(Early Start Schedule)이 있다[7]. 늦은 작업계획은 작업자가 유휴 시간없이 항상 연속적으로 작업할 수 있도록 작업을 계획하는 방법이고, 빠른 작업계획은 각 작업장에 기준점(Reference

Point)을 두어 이 위치 앞으로 이동하여 작업을 할 수 없도록 계획하는 방법이다. 늦은 작업계획은 빠른 작업계획에 비해 상대적으로 라인길이가 길 수 있으며, 반면에 빠른 작업계획은 작업자의 유휴 시간을 허용함으로써 생산시간이 손실될 수 있다[3]. 본 연구에서는 빠른 작업계획을 사용한다. 둘째, 혼합모델 조립라인에서 계획기간(Planning Horizon) 동안 M 종류의 모델이 생산된다고 하자. 그리고 각 모델의 생산개수를 각각 D_1, \dots, D_M 이라고 하고, D_1, \dots, D_M 의 최대공약수를 h 라 하자. 각 모델의 수요 $(D_1/h, \dots, D_M/h) = (d_1, \dots, d_M)$ 를 최소 부품집합 (Minimum Part Set : MPS)이라고 한다[3]. 최소 부품 집합은 한 생산사이클 동안 생산되는 모델 수의 집합을 나타낸다. 혼합모델 조립라인에서는 MPS를 h 번 반복 생산함으로써 계획기간의 수요를 만족시키는 사이클 생산을 흔히 사용하고 있다. 한 사이클 동안 생산되는 제품의 총 수, I 는 $\sum_{m=1}^M d_m$ 이 된다. 셋째, 제품의 투입시간 간격(γ)은 $\gamma_1 + \gamma_2$ 로 둔다. 여기서 γ_1 은 각 작업장당 모델의 평균 조립작업시간을 나타낸다. 그리고 t_{jm} 은 작업장 j 에서 모델 m 의 조립시간으로, s_{jmr} 은 작업장 j 에서 모델 m 에서 모델 r 로의 변경에 따른 준비시간으로 두자. γ_1 은 1회의 MPS를 생산하는 데 요구되는 총 조립시간($T = \sum_{j=1}^J \sum_{m=1}^M t_{jm} d_m$)을 제품의 총수, I 와 작업장의 수, J 의 곱으로 나눈 값이 된다. 또한 γ_2 는 각 작업장당 모델의 변화에 따른 평균 준비시간을 나타내며, 여기서는 전체 작업장에서 준비시간이 발생할 수 있는 모든 경우에 대한 총 준비시간($S = \sum_{j=1}^J \sum_{m=1}^M \sum_{r=1}^M s_{jmr} d_m d_r$)을 모델변화의 모든 경우의 수, $\beta_2^{\text{avr}} (= I \times (I - 1))$ 와, 작업장의 수 J

의 곱으로 나눈 값으로 둔다.

그림 1은 A, C, B, A, A, C 모델순으로 투입하는 경우, 작업장 j에서 각 모델의 작업 시작 위치와 종료 위치를 나타내고 있다. 그림에서 굵은 실선은 각 조립작업에 필요한 작업장길이를 나타내며, 가는 실선은 다음 제품의 준비시간동안 움직인 컨베이어의 이동거리를 나타낸다. 또한 점선은 제품간의 간격(w)을 나타내고, 화살표로 표시된 실선은 작업자의 유휴시간 동안 움직인 컨베이어의 이동거리를 나타낸다. 여기서 제품간의 간격은 제품의 투입시간 간격과 컨베이어의 속도를 곱하면 된다. 작업자는 각 작업장의 기준점에서 사이클 생산의 첫번째 모델의 작업을 시작한다. 컨베이어 상에서 모델의 조립작업을 완료한 후 컨베이어 앞으로 이동하여, 다음 모델이 앞 모델과 다르면 작업준비를 한다. 작업준비는 컨베이어 밖에서 이루어진다. 작업자가 준비작업을 마친 시점에 다음 제품이 작업장 경계(기준점)내에 도달해 있다면 조립작업을 실시하고(두번째 제품의 모델 C), 그렇지 않다면 작업자는 제품이 작업장에 도

달할 때까지 유휴하게 된다(네번째 제품의 모델 A).

작업장에 할당된 작업을 작업장 경계 내에서 완수하지 못하면 컨베이어 정지가 발생하게 된다. 따라서 컨베이어 정지위험은 작업장의 실제 길이(L_j)와 작업 수행에 필요한 작업장 길이(Y_j)를 사용하여 나타낼 수 있다[18]. 여기서 필요한 작업장 길이는 기준점에서부터 가장 늦은 작업완료 위치까지의 길이를 의미한다. 작업장 j의 여유길이는 작업장의 실제 길이(L_j)와 가장 늦은 작업의 종료위치(Y_j)의 차, $L_j - Y_j$ 로 정의할 수 있다. 따라서 작업장의 여유길이가 작을수록 컨베이어 정지위험은 높아지게 된다. 즉, 컨베이어 정지위험을 최소로 하는 투입순서는 모든 작업장의 여유길이 중 최소의 작업장 여유길이를 최대로 하는 투입순서를 구하면 된다. 또한 작업장에 대한 실제 길이(L_j)를 상수로 두면, 조립작업을 완수하기 위해 필요한 작업장의 길이(Y_j) 중 가장 큰 것을 최소로 하는 투입순서를 구함으로써 컨베이어 정지위험을 최소로 할 수 있다. 본 연구에서는 작업장에 대한 실제 길이(L_j)를 상수로 두고 접근하였으며, 작업장의 중요도를 조정할 수 있도록 가중치를 사용하였다. 즉 목적 1은 식 (1)과 같이 둔다.

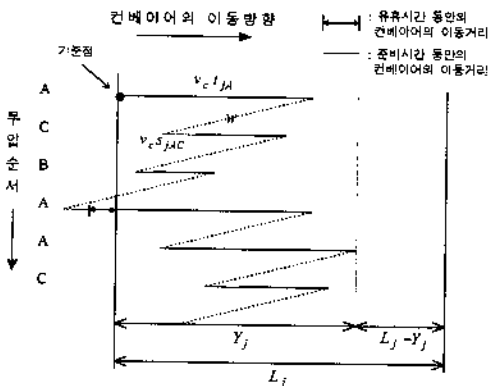


그림 1. 작업장 j에서의 작업장의 실제 길이와 각 작업의 종료 위치

$$\text{목적 1: } \min_j \max_j \alpha_j Y_j \quad (1)$$

혼합모델 조립라인에서 부품사용의 일정률을 유지하기 위한 목적은 제품생산에 따른 각 모델별 실제 생산 비율과 수요량 비율의 차를 최소로 함으로써 달성할 수 있다[17]. 본 연구에서 부품 사용의 일정률을 유지하기 위

한 목적은 Miltenberg[16]가 제시한 목적함수를 사용한다. 이는 식(2)와 같다.

$$\text{목적 2: } \text{Min. } \sum_{i=1}^I \sum_{m=1}^M \left| \sum_{r=1}^I \sum_{i=1}^M \frac{X_{lmr}}{i} - \frac{d_m}{I} \right| \quad (2)$$

여기서 X_{lmr} 은 투입순서에서 모델 m 이 l 번째에 위치하고, 모델 r 이 $(l+1)$ 번째에 위치하면 1이고, 그렇지 않으면 0인 변수이다. 목적 2에서 절대값 안의 첫번째 항은 i 번째 투입 순서까지 생산한 모델의 생산 비율을 나타내고, 두 번째 항은 수요 비율을 나타낸다.

부품의 사용량을 일정하게 유지하는 모델의 투입순서는 완제품 및 재공품 재고를 가능한 한 최소로 유지시키면서 제품의 수요를 만족시키고자 하는 JIT 생산시스템에 잘 부합된다고 알려져 있다[16, 17]. 다루는 투입 순서 문제의 수리모형은 현철주의 논문[2]을 참조할 수 있다.

3. 다목적 유전알고리즘

유전알고리즘은 해를 표현하는 개체로 구성된 모집단을 변화시키면서 최적해를 탐색해 간다. 모집단의 각 개체를 평가함수에 의해 평가하여, 이를 기초로 적합한 개체를 선별함으로써 새로운 모집단을 형성한다. 새로 형성된 모집단의 개체를 교차변이 또는 돌연변이를 시켜 새로운 개체를 재생산한다. 이 과정을 종료조건이 만족할 때까지 반복하게 된다.

3.1 기존의 다목적 유전알고리즘

유전알고리즘을 이용하여 다목적 문제의 파레토 최적해를 찾기 위한 연구는 1984년

Schaffer[20]에 의해 시작되었다. Schaffer[20]는 다목적 문제에서 다수의 해를 구하기 위해 Vector Evaluated GA(VEGA)를 제시하였다. 다목적 문제는 하나의 개체에 여러 개의 목적함수 값이 대응되므로 다목적 문제의 목적함수 값은 벡터 형태를 이룬다. VEGA는 모집단을 목적의 개수 만큼의 하위 그룹으로 나누고, 각 하위 그룹에 각 목적을 대응시켜, 각 하위 그룹에서 그 목적에 적합한 개체들의 선별 확률을 높게 하여, 파레토 최적해를 찾는다. VEGA는 쉽게 구현가능하고 극점 부근에 위치한 파레토 최적해는 잘 찾는 반면에, 중간 부근에 위치한 파레토 최적해는 찾기가 어려운 것으로 알려져 있다[8, 19].

Goldberg[10]는 VEGA의 단점을 보완하여 비지배 순위를 이용한 Pareto GA를 제시하였다. Horn *et al.*[12]은 다양한 파레토 최적해를 찾기 위해 이진 토너먼트(Binary Tournament) 선별방법을 변형하고 적소의 개념을 도입한 Niche Pareto GA를 제시하였다. Niche Pareto GA는 2개의 후보개체와 g 개의 개체로 구성된 비교집합을 임의로 선택한 후, 2개의 후보개체를 비교집합내의 각 개체와 비교한다. 만일 하나의 후보개체는 비교집합의 모든 개체를 지배하고, 다른 후보개체는 비교집합의 어떤 개체에 의해 지배된다면, 전자의 후보개체를 선별한다. 한편 두 후보 모두 비교집합의 모든 개체에 지배되지 않거나, 비교집합의 어떤 개체에 의해 지배되면, 두 후보 개체에 대해 적소의 개념을 사용하여 개체를 선별한다. 본 연구에서는 각 개체의 비지배 순위와 적소의 개념을 이용하여 선별을 결정할 수 있는 순위-적소 유전알고리즘을 제시한다.

3.2 순위-적소 유전알고리즘

순위-적소 유전알고리즘은 비지배 순위와 적소의 개념을 결합하여 각 개체의 선별을 결정할 수 있도록 구현한 다목적 문제를 위한 유전알고리즘이다.

선별계획(Selection Scheme)은 각 개체의 평가함수 값을 기초로 다음 세대의 생존개체를 선택하는 방법으로, 좋은 해가 갖는 정보의 효율적인 이용(Exploitation)과 해공간을 효율적으로 다양하게 탐색(Exploration)하는 데 커다란 영향을 미치는 요소이다. 선별방법에는 비례적 재생산(Proportionate Reproduction), 토너먼트 선별(Tournament Selection), 순위 선별(Ranking Selection) 등이 있다[9].

순위-적소 유전알고리즘은 각 개체의 순위에 따라 생존 확률을 부여하는 순위선별을 기본으로 하고 있다. 그러나 다목적 문제는 한 개체가 여러 개의 목적함수 값을 가지고 있으므로 각 개체의 순위를 결정하기 위해 평가함수 값의 순위가 아닌 비지배 순위를 사용하였다. 또한 순위-적소 유전알고리즘에서는 비지배 순위만을 이용한 Pareto GA와 달리 하나의 개체로 수렴을 피하고 다양한 파레토 최적해를 유지하기 위하여 적소의 개념을 도입하였다. 그리고 적소영역의 결정방법 및 적소의 개념을 선별방법에 결합하는 방법을 제시하였다.

3.2.1 비지배순위

앞에서 언급하였듯이 유전알고리즘을 다목적 문제에 적용하는 경우, 한 개체가 다수 개의 평가함수 값을 가지고 있으므로 평가함수 값을 직접적으로 사용하여 개체를 선별하는 데는 어려운 점이 많다. 따라서 다목적 문제

는 비지배 순위를 이용하여 개체들을 선별하고 있다[8, 10]. 본 연구에서는 Goldberg[10]가 제시한 비지배 순위방법을 사용한다. 비지배 순위의 결정은 다음과 같다. 현재 모집단에서 비지배되는 개체들을 찾는다. 비지배되는 개체들을 순위 1로 두고 이들을 제외한 후, 다시 모집단에서 비지배되는 개체들을 찾아 순위 2로 두고 이들을 제외한 후, 다시 비지배되는 개체를 찾아 순위 3으로 둔다. 이와 같은 과정을 모집단의 모든 개체들이 순위가 부여될 때까지 반복한다. 이는 비지배적 순위가 좋은 개체일수록 선별될 확률을 높게 두고자 함이다.

3.2.2 適所

적소의 개념은 적합한 곳에 위치한 개체들이 선별될 확률을 보다 높게 두자는 것으로, Multi-modal 문제에서 다수의 해를 찾기 위해 고안된 개념이다[10]. Multi-modal 문제에서는 각 개체들의 적소정도를 파악하여 이를 평가함수값에 반영한 공유 평가함수값(Shared Fitness)을 이용하여 선별을 결정한다. 그러나 다목적 문제에서는 한 개체가 다수의 목적함수값을 가지기 때문에 Multi-modal 문제에서 사용한 방법을 그대로 적용하기는 어렵다. 다목적 문제에 적소의 개념을 이용하기 위해서는 먼저 적소영역이 정의되어야 하고, 또한 이 적소영역으로 부터 얻은 정보를 선별에 이용하는 방법이 고안되어야 한다.

본 연구에서 적소영역은 현 세대의 개체들이 분포되어 있는 영역을 전체 개체의 수로 나눈 영역을 사용하였다. n 개의 목적을 갖고 있는 다목적 문제에서 어떤 세대마다 l 번째 목적함수의 최대값과 최소값을 각각 MAX_l ,

$MIN_l, l = 1, 2, \dots, n$ 이라고 하자. 그러면 적소영역을 구성하는 $\sigma_l, l = 1, 2, \dots, n$ 는 모집단의 크기(pop_size)와 $MAX_l, MIN_l, l = 1, 2, \dots, n$ 에 의해 식 (3)과 같이 결정된다.

$$\sigma_l = \frac{MAX_l - MIN_l}{n \cdot pop_size}, l = 1, \dots, n \quad (3)$$

매 세대마다 식 (3)과 같이 적소영역 σ_l 을 구하여 사용한다.

그림 2는 두 개의 목적을 가진 문제에서 모집단의 크기가 25일 때 적소영역을 결정한 예를 나타낸다.

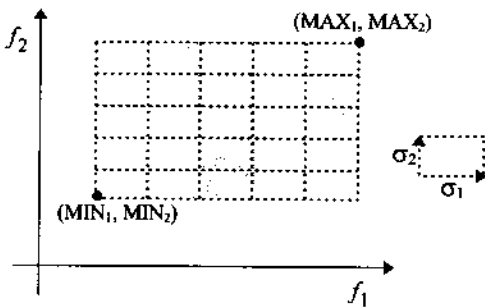


그림 2. 적소영역의 결정 (모집단 크기 = 25)

각 개체의 선별에 적소의 개념을 도입하기 위해 적소영역내에 존재하는 개체들의 수 (Niche Count)를 사용하였다. 각 개체의 선별 확률을 같게 두면, 세대가 진행됨에 따라 Niche Count가 적은 영역에 개체가 존재할 확률은 아주 낮아지게 된다. 따라서 다양한 해들을 생존케 하여 궁극적으로 다양한 파레토 최적해를 구하기 위하여, Niche Count가 큰 영역내의 개체보다 적은 영역 내에 있는 개체의 선별 확률을 크게 두었다.

비지배 순위와 적소의 개념을 이용한 순위-적소 유전알고리즘의 선별절차를 단계별

로 나타내면 다음과 같다.

단계 1: 각 개체의 비지배 순위와 적소영역내의 개체 수를 파악하여 모집단의 각 개체에 순위를 준다.

현재 모집단에서 비지배되는 개체들을 찾는다. 비지배되는 모든 개체에 대해 적소영역내의 개체 수를 계산한다. 적소영역내의 개체 수가 가장 작은 개체부터 차례로 나열하여 순위를 1, 2, ..., N_i 로 준다. 이들을 제외한 후, 다시 모집단에서 지배당하지 않는 개체들을 찾는다. 다시 이들에 대한 적소영역내의 개체수를 구하고, 적소영역내의 개체수가 가장 작은 개체부터 차례로 나열하여 순위를 N_i+1, N_i+2, \dots, N_i 로 준다. 이 과정을 모집단의 모든 개체들이 순위를 가질 때까지 반복한다.

단계 2: 각 개체가 선별될 확률은 다음과 같은 기하분포(Geometric Distribution)로 준다.

$$Prob(rank) = q(1-q)^{rank-1}, rank = 1, 2, 3, \dots, pop_size$$

여기서 q 는 각 개체의 선별확률 파라미터로, $0 < q < 1$ 의 값을 갖는다.

이러한 선별전략은 비지배 순위가 높은 개체의 선별확률을 크게 하여 좋은 해를 생산하고, 비지배 순위가 동일한 경우 적소의 위치에 있는 개체의 선별확률을 크게 줌으로써 다양한 비지배 해를 얻고자 한다.

3.3 순위-적소 선별방법의 적용을 위한 유전알고리즘 설계

순위-적소 선별방법의 적용을 위해서는 표현방법, 평가함수, 모집단의 초기화 방법, 유전연산자 및 유전파라미터의 결정 등이 이루

어져야 한다. 본 절에서는 순위-적소 선별방법이 적용되는 유전알고리즘의 요소에 대해 살펴보기로 한다.

3.3.1 수정 유전알고리즘(modified Genetic Algorithm : modGA)

Holland[11]에 의해 최초로 제시된 유전알고리즘으로부터 해의 효율 및 계산시간의 단축을 위해 이를 변형한 여러 유전알고리즘이 개발되었다. 본 연구에서는 높은 평가함수값을 갖는 초개체가 다음 세대에 여러개 나타나는 것을 방지하고 돌연변이의 처리를 개체단위로 처리할 수 있는 수정 유전알고리즘[15]을 기본으로 하고자 한다. 수정 유전알고리즘의 절차는 다음과 같다. 여기서 $P(t)$ 는 세대 t 에서의 모집단을 의미한다.

단계 1: 초기모집단($P(t=0)$)을 생성하고 평가한다.

단계 2: 모집단 $P(t)$ 에서 k 의 개체를 선택한다.

단계 3: 단계 2에서 선택된 개체들을 교차변이 또는 돌연변이를 시켜 새로운 k 개의 개체를 생산한다.

단계 4: 생산된 k 의 개체를 모집단 $P(t+1)$ 에 복사한다.

단계 5: 모집단 $P(t)$ 에서 $pop_size - k$ 의 서로 다른 개체를 선택하여 모집단 $P(t+1)$ 에 복사한다.

단계 6: 모집단 $P(t+1)$ 을 평가한다.

단계 7: 종료조건을 만족하면 알고리즘의 수행을 마치고, 그렇지 않으면 $t \leftarrow t+1$ 로 두고 단계 2로 간다.

3.3.2 표현방법

문제의 해를 개체로 표현하는 방법은 문제의 특성에 따라 다르나, 유전연산자를 사용하여 가능해 공간을 효율적으로 탐색할 수 있도록 표현되어야 한다. 본 연구에서는 자연스럽고 중복(Redundancy)이 없는 표현방법으로 사이클 생산내에서 생산되는 모델들의 목록을 투입순서대로 나열하는 방법을 사용하고자 한다. 예로 사이클 생산 동안 모델 A, B, C가 각각 2, 3, 4개 생산되어야 한다고 하자. 그리고 하나의 가능해로 투입순서가 A, C, B, C, B, A, C, C, B라면 개체의 표현방법은 모델의 투입순서대로(A C B C B A C C B)와 같이 표현한다.

3.3.3 초기 모집단과 평가함수

본 연구에서는 초기모집단으로 사이클생산 동안 각 모델의 수요량을 만족하는 임의의 해를 생성하여 사용하고, 평가함수로는 제 2장의 목적함수 식(1)과 (2)을 사용한다. 평가함수에 의해 모집단을 구성하는 각 개체들이 평가되며, 이러한 평가된 값(Fitness)을 기초로 하여 다음 세대의 개체들을 재생산한다. 다루는 문제는 2개의 목적을 가지고 있으므로, 각 개체의 평가함수값은 2개를 갖게 된다.

3.3.4 유전연산자

순서문제에서는 교차변이와 돌연변이 대신에 두 부모를 결합하여 자손을 생산하는 이원연산자와 한 부모의 유전인자들을 변화시켜 자손을 생산하는 일원연산자로 흔히 나누고 있다. 본 연구에서는 이원연산자로는 인자들의 직후행관계의 정보를 이용하여 자손

을 생산하는 ISRX(Immediate Successor Relationship Crossover) 연산자를, 일원연산자로는 역순연산자를 사용하기로 한다[1]. 이들 연산자는 혼합모델 투입순서문제에서 좋은 성능을 나타내었다[1, 14]. ISRX 연산자는 인자별 직후행 인자와 인자비율을 이용하여 자손을 생산한다. 부모1(P1)과 부모2(P2)가 각각 (A A B B B C C C C), (A B C A B C B C C)일 때, 두 부모에서 각 인자에 따른 직후행인자와 인자비율을 표 1(a)에 나타내었다. 표 1(a)에서 인자형 A의 직후행인자는 P1에서 A, B이고, P2에서 B, B가 되어 A, B, B, B가 되며, 다른 인자형의 경우에도 같은 방법으로 결정된다. 또한 인자비율은 자손에게 아직 상속되지 않는 직후행인자들의 수에 대한 비율로 표 1(a)에서 A, B, C인자들의 직후행인자의 개수는 각각 4, 6, 8개이므로 인자비율은 2:3:4가 된다.

ISRX 연산자가 자손을 생산하는 절차는 다음과 같다.

단계 1: 두 부모로부터 인자별 직후행인자와 인자비율을 구한다. 즉, ISRX 표를 만든다.

단계 2: $i = 1$ 로 두고 임의로 인자형 e_i 를 선택한다.

단계 3: 자손의 i 번째 위치에 e_i 를 유전시킨다. 만일 하나의 자손이 완전히 생산되었으면 끝내고, 아니면 다음 단계로 간다.

단계 4: 직후행인자의 열에서 인자형 e_i 의 원소 2개를 임의로 삭제하고, 인자비율을 수정한다.

단계 5: i 를 1 증가시키고, 단계 4에서 삭제된 인자형의 행에서 가장 많이 남아있는 인자형을 선택한다. 동일한 경우 인자비율이 작은 것을 선택하고, 이것 또한 동일한 경우 임의로 선택한다. 선택된 인자를 e_i 로 둔다. 단계 3으로 간다.

위의 방법에서, 단계 4에서 인자 2개를 임의로 삭제하는 것은 두 부모의 직후행인자를 모두 고려하였기 때문이다. 또한 단계 5와 같이 인자를 선택하는 것은 부모의 직후행 인자관계를 자손에게 상속하고자 하는 것이다.

ISRX 연산자의 예는 다음과 같다. 위 예에서, 표 1(a)과 같이 ISRX표를 만든 후, 시작인자로 B를 임의 선택하였다고 하자. 선택된 인자 B를 자손의 첫번째 위치에 유전시킨 후,

표 1. ISRX 표

인자형	(a)		(b)		(c)		(d)	
	직후행 인자	인자 비율	직후행 인자	인자 비율	직후행 인자	인자 비율	직후행 인자	인자 비율
A	ABBB	2	ABB	2	ABB	2	ABB	1
B	BBCCCC	3	BCCCC	2	BCCC	2	BCCC	2
C	CCCAABCA	4	CCCAABCA	4	CCAABCA	3	CCBCA	3
생산된 인자	B		C		A		B	

ISRX표의 직후행 인자열에서 임의로 2개의 B를 삭제한다. 인자형 A와 B에 있는 직후행 인자 B를 삭제하면, 표 1(b)가 된다. 선택된 B인자와의 직후행 관계는 ISRX표의 B인자형에서 B가 1개, C가 4개로 C가 가장 밀접하므로 C를 선택하여 자손의 두번째 위치에 유전시킨다. 다시 ISRX표의 직후행 인자열에서 임의로 2개의 C를 삭제한다. 인자형 B와 C에 있는 C를 삭제하면, 표 1(c)가 된다. 인자형 C에 남아 있는 갯수는 A와 C가 3개로 동일하나, 그 인자비율이 A:C=2:3이므로 A를 선택하여 자손의 세번째 위치에 유전시킨다. 이와 같이 반복하면 (B C A B C C B C A)와 같은 자손이 생산될 수 있다. 다른 자손의 경우에는 시작인자를 다시 임의선택하여 위와 같은 단계를 따른다.

역순연산자는 한 부모에서 두 개의 교차점을 임의로 선택하고 교차점사이의 인자들을 역순으로 하여 자손을 생산하는 일원연산자이다. 예로 그림 3의 부모(P)에서 서로 다른 두 교차점이 임의로 선택된 경우, 아래와 같이 교차점사이의 인자들을 역순으로 하여 자손(O)이 생산된다.

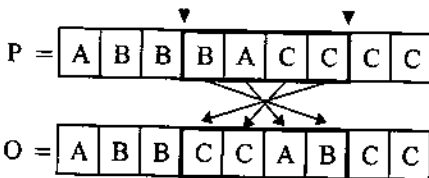


그림 3. 역순연산자

4. 실험 및 결과 분석

순위-적소 유전알고리즘의 성능을 분석하기 위하여 유전파라미터 값을 실험을 통하여

결정하고, 기존의 다목적 유전알고리즘인 VEGA, Pareto GA, Niched Pareto GA와 제한한 순위-적소 유전알고리즘의 성능을 비교분석하고자 한다.

사용된 모든 유전알고리즘은 C++ 언어로 프로그래밍되어 Pentium CPU를 가진 IBM-PC를 사용하여 실험하였다.

4.1 실험예제

예제에서 다루는 혼합모델 조립라인은 총 12개의 폐쇄작업장으로 구성되어 있으며, 모든 작업장에서 모델의 변화에 따라 준비시간이 발생한다고 본다. 또한 컨베이어의 속도 (v_c)와 작업장 j 의 중요도를 나타내는 α_j 는 계산상의 편의를 위해 모두 1로 두었다. 작업장 j 에서 모델 m 에 대한 작업시간(t_{jm})은 $20 \leq t_{jm} \leq 40$, 모든 j, m , 범위를 갖는 일양분포(Uniform Distribution)에서 발생하였다. 이러한 상황에서 다양한 문제를 통해 제한한 유전알고리즘의 성능을 분석하기 위하여 총 제품수(I)와 준비시간의 수준에 따라 문제를 분류하였다.

표 2는 생산할 총 제품 수를 3수준, 준비시간분포를 4수준으로 분류한 12개의 예제문제를 나타내고 있다. 준비시간분포에서 0은 준비시간이 없음을, $U(a, b)$ 는 $a \leq s_{jmr} \leq b$, 모든 $j, m \neq r$ 범위를 갖는 일양분포에서 준비시간을 발생하였음을 의미한다. 또한 표 2의 비교에서 생산제품의 수에 따른 모델의 수와 모델별 생산제품수(MPS)를 나타내었다. 문제 1의 경우, 10종류의 모델에 대해 생산할 총 제품수는 20개이므로 한 개체의 인자수는 20개이며, 한 개체에서 모델별 생산제품수 만큼 같은 인자가 나타나게 된다.

표 2. 유전알고리즘의 성능평가를 위한 문제 설계

문 제	생산제품의 총수	준비시간의 분포	비 고	
			모델의 수	모델별 생산제품수(MPS)
S - 1	20	0	10	(2,1,3,1,4,3,1,3,1,1)
S - 2	20	U(1, 3)	10	(2,1,3,1,4,3,1,3,1,1)
S - 3	20	U(1, 7)	10	(2,1,3,1,4,3,1,3,1,1)
S - 4	20	U(1, 15)	10	(2,1,3,1,4,3,1,3,1,1)
M - 1	30	0	15	(2,1,3,1,5,3,1,4,1,2,2,1,2,1,1)
M - 2	30	U(1, 3)	15	(2,1,3,1,5,3,1,4,1,2,2,1,2,1,1)
M - 3	30	U(1, 7)	15	(2,1,3,1,5,3,1,4,1,2,2,1,2,1,1)
M - 4	30	U(1, 15)	15	(2,1,3,1,5,3,1,4,1,2,2,1,2,1,1)
L - 1	40	0	20	(3,1,4,3,5,2,1,2,1,2,1,2,1,2,2,1,1,3,2,1)
L - 2	40	U(1, 3)	20	(3,1,4,3,5,2,1,2,1,2,1,2,1,2,2,1,1,3,2,1)
L - 3	40	U(1, 7)	20	(3,1,4,3,5,2,1,2,1,2,1,2,1,2,2,1,1,3,2,1)
L - 4	40	U(1, 15)	20	(3,1,4,3,5,2,1,2,1,2,1,2,1,2,2,1,1,3,2,1)

4.2 유전파라미터 값의 결정

유전파라미터는 모집단의 크기(*pop_size*), 이원연산자와 일원연산자의 비율, 연산자 적용비율(*k*), 선별확률파라미터(*q*), 그리고 종료조건이 있다. 유전파라미터의 값을 구하는 이론 연구는 아직 미흡하다[12]. 따라서 실험을 통하여 이들 파라미터 값을 결정하였다.

첫째, 다루는 문제에서 모집단의 적정크기를 분석하기 위해 100, 200, ..., 1000으로 변화시키면서 실험하였다. 모집단의 크기가 클수록 좋은 비지배해를 다양하게 찾을 수 있었으나, 계산시간은 증가하게 된다. 따라서 이들 관계의 분석을 통하여 모집단의 크기를 300으로 두었다.

둘째, 매 세대마다 유전연산에 의해 새로 생산되는 개체들의 수(*k*)는 모집단 크기의

50%로 두었다.

셋째, 이원연산자와 일원연산자의 비율은 이원연산자인 ISRX와 일원연산자인 역순연산자를 3 : 7로 혼용하여 사용하였다. 이는 연산자의 적용개체수(*k*)의 개체 중 30%가 이원연산자에 의해 생산되고, 나머지 70%가 일원연산자에 의해 생산된다는 것을 의미한다.

넷째, 수렴의 속도를 조절하는 역할을 하는 선별확률파라미터(*q*)의 값에 대해 실험한 결과, $5/pop_size$ 에서 $9/pop_size$ 사이의 값이 비교적 만족할 만한 해를 얻었다. 본 연구에서는 *q*의 값을 0.02, 즉 각 개체의 평균선별확률($1/pop_size = 1/300$)의 6배로 두기로 한다. 또한 우수개체(Elitism) 선별전략을 변형하여 한 세대에서 나타난 모든 비지배해를 다음 세대에 유지하도록 하였다.

다섯째, 알고리즘의 종료조건으로 세대수를 사용하였고, 한 사이클당 생산제품의 수가 20, 30, 40 인 경우, 각각 세대수가 100, 150, 200에 이르면 알고리즘을 종료하였다. 이러한 종료조건은 각 문제에서 해의 향상 정도와 계산시간에서 적절하였다.

4.3 VEGA, Pareto GA, Niche Pareto GA, 순위-적소 유전알고리즘의 비교

Schaffer[20]가 제시한 VEGA, Goldberg[10]가 제시한 Pareto GA, Horn *et al.*[12]이 제시한 Niche Pareto GA, 그리고 본 연구에서 제시한 순위-적소 유전알고리즘을 해의 다양성 및 효율성, 그리고 계산시간 측면에서 비교 분석하고자 한다.

VEGA는 모집단을 2개의 그룹으로 나누고, 각 그룹마다 각 목적에 적합한 개체들이 선별될 확률을 높게 하였다. 즉, 한 그룹은 목적 1의 값이 가장 좋은 개체부터 차례로 나열하여 순위를 주고, 각 개체가 선별될 확률은 기하분포를 사용하여 재생산하였고, 다른 그룹은 목적 2의 순위에 따라 같은 방법으로 재생산하였다. Pareto GA는 순위-적소 유전알고리즘에서 적소를 사용하지 않고, 비지배 순위만을 이용하였다. Niche Pareto GA에서 사용한 적소영역은 3.2.2절에서 제시한 방법으로 적정한 비교집합의 개체수(g)를 결정하기 위해 5, 10, ..., 50으로 변화시키면서 실험하여 비교적 좋은 결과를 얻은 20을 사용하였다.

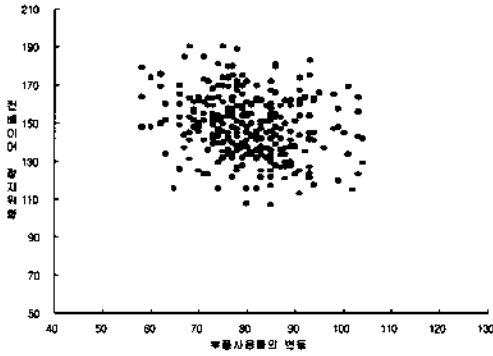
다목적 유전알고리즘의 해의 탐색성능과 분포를 살펴보기 위해, 문제 S-3으로 실험하여 그림 4에 나타내었다. 그림 4(a)는 임의로 발생한 300개의 개체로 구성된 초기 모집단

분포를 나타내며, (b),(c),(d),(e)는 각 알고리즘에 따라 300세대 진행한 후의 모집단 분포를 나타내고 있다. 그림 4에서 x축과 y축은 각각 목적함수 1과 2의 값을 나타낸다.

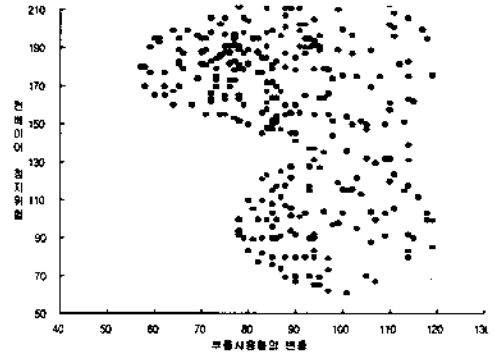
또한 다목적 유전알고리즘에 대한 비교분석의 신뢰성을 높이기 위해 각 문제마다 초기해를 달리하여 30회를 반복실험하고, 이들의 평균치로 나타내었다. 표 3(a)는 VEGA, Pareto GA(PGA), Niche Pareto GA(NPGA), 순위-적소 유전알고리즘을 사용하여 탐색한 비지배 해의 개수를 나타내며, (b)는 이들 다목적 유전알고리즘으로 구한 모든 비지배 해들의 집합에서 다시 비지배 해를 구한 후, 이들이 어떤 방법에 의해 탐색된 비지배 해인지를 구별하여 그 개수를 나타내었다.

그림 4에서 VEGA는 각 목적에 적합한 개체를 독립적으로 선택하기 때문에 극점근처가 아닌 중간부분에 위치한 파레토 최적해를 찾기가 어렵다는 것을 알 수 있다. 그림 4와 표 3(a)로 제안한 순위-적소알고리즘은 비교한 세 유전알고리즘과 보다 다양한 비지배해를 찾으며, 표 3(b)로부터 가장 좋은 비지배해를 구한다는 것을 알 수 있다. 또한 Pareto GA와 Niche Pareto GA를 비교할 때, 표 3으로부터 해의 다양성 측면에서는 거의 같은 수준의 성능을 보였으나 좋은 비지배해를 찾는 데는 Pareto GA가 더 우수하였다. VEGA는 해의 다양성과 좋은 비지배해를 찾는 데 있어서 모두 가장 성능이 좋지 않았다.

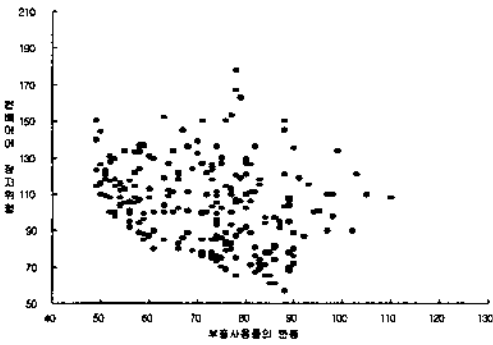
또한 표 3(b)에서 모델변화에 따른 준비시간이 없는 경우에는 1-2개의 비지배해만이 존재하나, 준비시간이 증가할수록 보다 많은 비지배해가 존재함을 알 수 있다. 이로부터 준비시간이 없는 경우에는 콰이어 정지위험



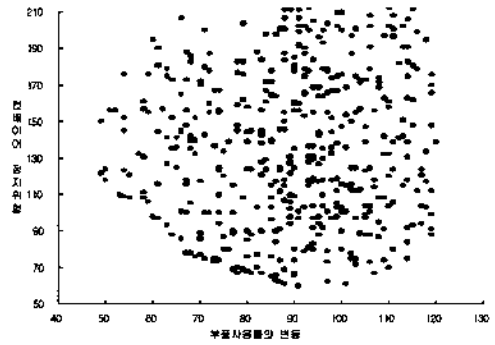
(a) 초기 모집단



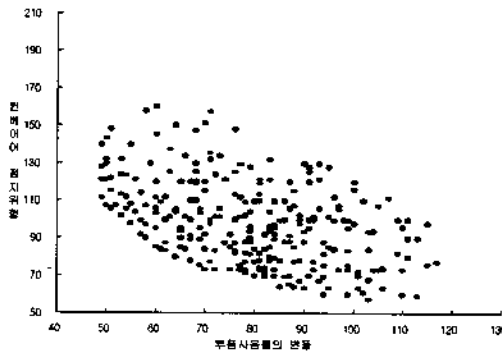
(b) VEGA



(c) Pareto GA



(d) Niched Pareto GA



(e) 순위-적소 유전알고리즘

그림 4. VEGA, Pareto GA, Niched Pareto GA, 순위-적소 유전알고리즘의 비교

의 최소화와 부품사용의 일정률 유지의 두 목적간에는 상충관계가 거의 없으나, 준비시간이 증가할 수록 두 목적간의 상충관계가

높게 나타난다는 것을 알 수 있다. 이러한 현상은 부품사용의 일정률을 유지하기 위해서는 각 모델의 수요량 비율에 따라 가능한 혼

표 3. 다목적 유전알고리즘의 해의 다양성 및 효율 비교

문제의 종류	(a)				(b)			
	VEGA	PGA	NPGA	순위적소	VEGA	PGA	NPGA	순위적소
S - 1	1.0	1.1	1.1	1.2*	0.7	1.0	0.9	1.0*
S - 2	11.3	13.7	14.1	15.1*	1.3	6.2	3.7	7.5*
S - 3	14.5	16.5	16.7	18.8*	2.0	7.7	3.9	8.4*
S - 4	18.6	19.8	19.5	21.7*	3.2	8.1	5.4	9.2*
M - 1	1.4	1.8	2.2*	2.1	0.8	1.0*	0.9	1.0*
M - 2	18.7	20.7	20.8	22.1*	2.9	6.7	3.6	10.8*
M - 3	19.8	22.6	23.1	25.5*	3.1	7.3	4.2	12.9*
M - 4	22.3	24.7	25.1	27.7*	3.4	9.4	5.1	14.1*
L - 1	1.3	1.4	1.4	1.5*	0.6	0.8	0.6	0.9*
L - 2	22.6	25.2	23.7	26.5*	3.3	9.5	5.2	12.8*
L - 3	23.5	26.8	26.9	28.1*	3.9	11.7	5.8	14.2*
L - 4	25.1	28.3	28.7	30.3*	4.4	12.3	5.8	14.7*

*는 비지배해의 개수가 비교대상 중 가장 많음을 나타냄

류생산하는 투입순서가 바람직하고, 컨베이어 정지위험의 목적도 준비시간이 적은 경우, 작업시간이 긴 모델과 짧은 모델을 교대로 투입함으로써 정지위험을 줄일 수 있어, 두 목적간의 상충관계가 약화되기 때문이다. 그러나 준비시간이 큰 경우에는 조립시간보다 준비시간에 상대적으로 영향을 많이 받아 각 모델을 로트생산하는 투입순서가 컨베이어 정지위험을 줄일 수 있기 때문에 두 목적간의 상충이 일어나는 것으로 풀이된다.

계산시간은 네 종류의 다목적 유전알고리즘에서 거의 차이가 없었다. 100세대 진행하는데 CPU시간은 네 알고리즘에서 S-1, 2, 3, 4문제는 약 220초, M-1, 2, 3, 4문제는 약 250초, L-1, 2, 3, 4문제는 약 290초가 소요되었

다.

순위-적소 유전알고리즘에서 세대수의 증가에 따른 비지배해의 변화과정을 분석하였다. 그림 5는 문제 L-3에 대한 세대수 증가에 따른 비지배해의 탐색과정을 나타낸다. 세대수가 진행될수록 보다 좋고 다양한 비지배해를 찾을 수 있다.

5. 결론

본 연구에서는 준비시간을 고려한 컨베이어 정지위험의 최소화와 부품사용의 일정을 유지를 목적으로 하는 혼합모델 조립라인의 투입순서 문제를 다루었다. 이 문제의 해결을 위하여 순위-적소 선별방법을 이용한 효

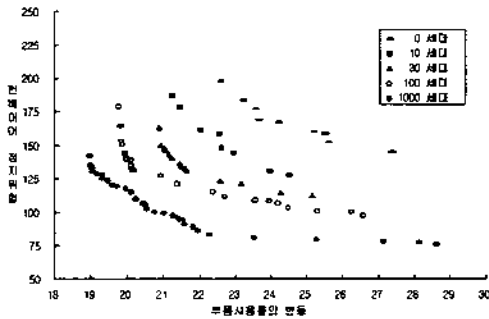


그림 5. 세대수 증가에 따른 비지배해 탐색과정

울적인 다목적 유전알고리즘을 제시하였다. 다양한 비지배해를 구하기 위하여 순위선별과 적소선별을 결합한 선별방법을 개발하였으며, 이를 기존의 다목적 유전알고리즘인 VEGA, Pareto GA, Niche Pareto GA와 비교 분석하였다. 실험결과, 제안된 유전알고리즘이 비교한 세 종류의 유전알고리즘보다 다양하고 좋은 비지배해를 찾았다. 또한 Pareto GA가 Niche Pareto GA보다 좋은 비지배해를 찾았으며, VEGA가 가장 좋지 않은 성능을 보였다.

참 고 문 헌

- [1] 김여근, 현철주, “혼합모델 조립라인의 생산순서결정을 위한 유전알고리즘”, 대한산업공학회지, 제 20권, 제 3호, pp. 15-34, 1994.
- [2] 현철주, “혼합모델 조립라인의 투입순서결정을 위한 유전알고리즘”, 박사학위논문, 전남대학교, 1996, 2.
- [3] Bard, J. F., Dar-El, E. M. and Shtub, A., “An analytic framework for sequencing mixed model assembly lines”, Int. J. Prod. Res., Vol. 30, pp. 35-48, 1992.
- [4] Bard, J. F., Shtub, A. and Joshi, S. B., “Sequencing mixed-model assembly lines to level parts usage and minimize line length”, Int. J. Prod. Res., Vol. 32, pp. 2431-2454, 1994.
- [5] Bolat, A. Savsar, M. and Al-Fawzan, M. A., “Algorithms for real-time scheduling of jobs on mixed model assembly lines”, Computers Ops Res., Vol. 21, pp. 487-498, 1994.
- [6] Burns, L. D. and Daganzo, C. F., “Assembly line job sequencing principles”, Int. J. Prod. Res., Vol. 25, pp. 71-99, 1987.
- [7] Dar-El, E. M., “Mixed-model assembly line sequencing problems”, Omega, Vol. 6, pp. 313-323, 1978.
- [8] Fonseca, C. M. and Fleming, P. J., “Genetic algorithms for multiobjective optimization: formulation, discussion and generalization”, Proc. 5th Int. Conf. Genetic Algorithms, pp. 416-423, 1993.
- [9] Goldberg, D. E. and Deb, K., “Comparative analysis of selection schemes used in genetic algorithms”, In Foundations of Genetic Algorithms(Edited by G. J. E. Rawlins), pp. 69-93. Kaufmann, 1991.
- [10] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, New York, 1989.
- [11] Holland, J. H., Adaptation in Natural and Artificial Systems, Univ. of Michigan Press, Ann Arbor, 1975.

- [12] Horn, J., Nafpliolas, N. and Goldberg, D. E., "A niched pareto genetic algorithm for multiobjective optimization", Proc. 1st IEEE Conference on Evolutionary Computation, pp. 82-87, 1994.
- [13] Kilbridge, M. and Wester, L., "The assembly line model-mix sequencing problem", Proc. 3rd Int. Conf. Op. Res., Oslo, Paris: English Universities Press, pp. 247-260, 1963.
- [14] Kim, Y. K., Hyun, C. J. and Kim, Y., "Sequencing in Mixed Model Assembly Lines : A Genetic Algorithm Approach", Computers & OR(in press), 1996.
- [15] Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer, Berlin, 1992.
- [16] Miltenburg, J., "Level schedules for mixed-model assembly lines in just-in-time production systems", Mgmt. Sci., Vol. 35, pp. 192-207, 1989.
- [17] Monden, Y., Toyota Production System, Institute of Industrial Engineers Press, Norcross, GA, 1983.
- [18] Okamura, K. and Yamashina, H., "A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor", Int. J. Prod. Res., Vol. 17, pp. 233-247, 1979.
- [19] Ritzel, B. J. and Eheart, J. W., "Using genetic algorithms to solve a multiple objective groundwater pollution containment problem", Water Resource Research, Vol. 30, pp. 1589-1603, 1994.
- [20] Schaffer, J. D., "Multiple objective optimization with vector evaluated genetic algorithms", Proc. 1st Int. Conf. Genetic Algorithms and Their Applications, pp. 93-100, 1985.
- [21] Tsai, L., "Mixed model Sequencing to Minimize Utility Work and the Risk of Conveyor Stoppage", Mgmt. Sci., Vol. 41, pp. 485-495, 1995.
- [22] Yano, C. A. and Rachamadugu, R., "Sequencing to minimize work overload in assembly lines with product options", Mgmt. Sci., Vol. 37, pp. 572-586, 1991.